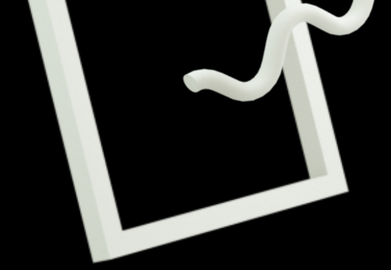
DevOps

EL HAKIK Amina



Why DevOps?

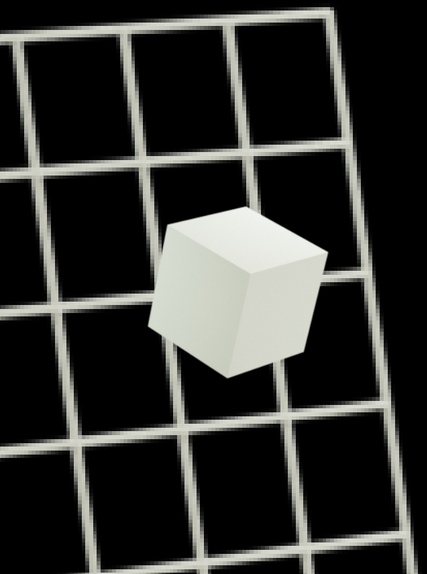
DevOps is necessary to address the challenges of modern software development, such as increasing deployment frequency, improving collaboration, achieving faster time-to-market, and ensuring reliable software delivery.

What is DevOps?

DevOps is a cultural and collaborative approach that brings together development and operations teams to streamline software development and deployment processes. It focuses on automation, continuous integration, continuous delivery, and close collaboration to enhance efficiency and quality.

DevOps benefits

DevOps offers several benefits, including faster delivery of features and bug fixes, reduced deployment failures, improved collaboration between teams, increased efficiency through automation, better scalability and resource utilization, and improved customer satisfaction.

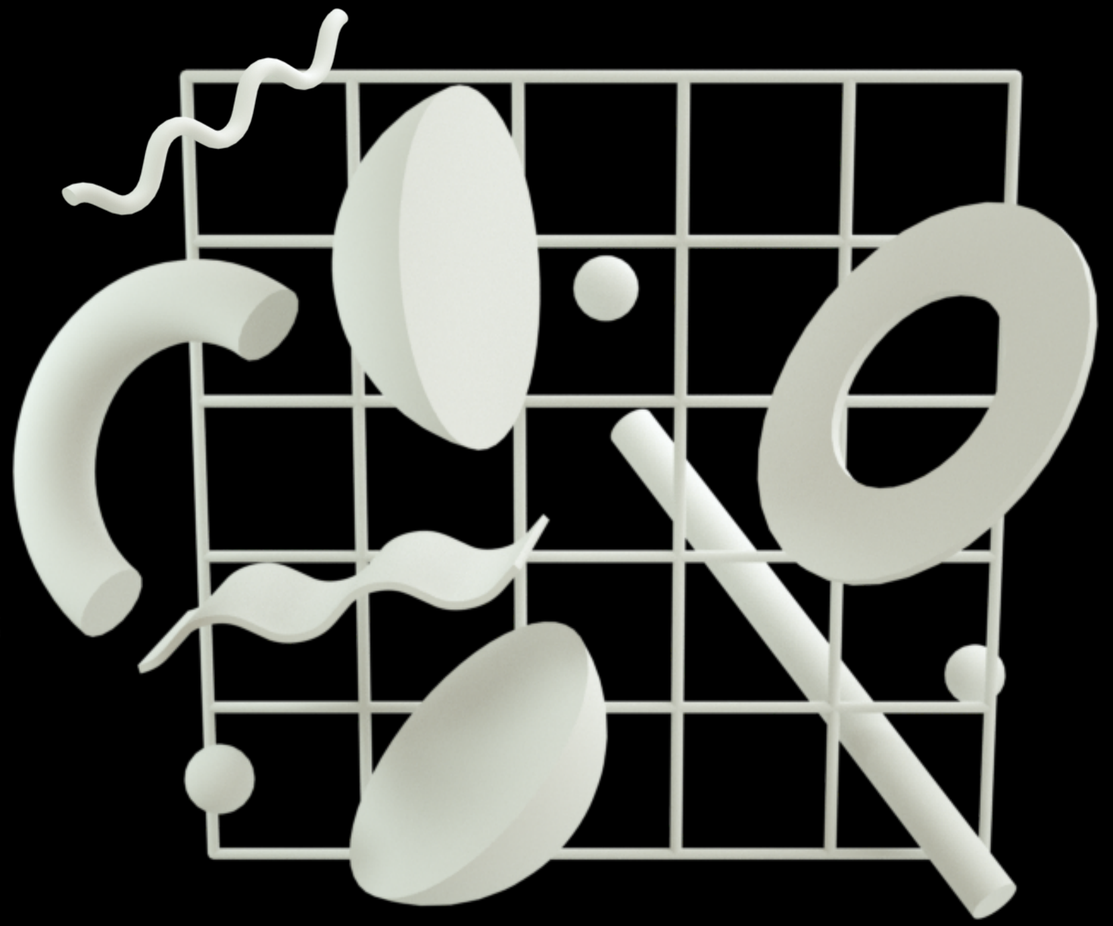


DevOps phases

DevOps typically involves different phases, including planning, coding, building, testing, deployment, monitoring, and feedback. These phases promote collaboration, automation, and continuous improvement throughout the software development lifecycle.

Who is a DevOps engineer?

A DevOps engineer is a professional responsible for bridging the gap between development and operations teams. They possess skills in software development, system administration, and automation tools. Their role involves implementing and managing DevOps practices, tools, and infrastructure to enable efficient and reliable software delivery.



What is version control?

Version control is a system that manages changes to files and code over time. It allows multiple contributors to work on the same project simultaneously, tracks changes, maintains a history of revisions, and facilitates collaboration and rollback to previous versions if needed.

What is Git?

Git is a widely used distributed version control system. It provides a reliable and efficient way to manage and track changes to files and code across multiple contributors. Git offers features such as branching, merging, and version history management.

Installing Git

To install Git, you can visit the official Git website (https://git-scm.com/) and download the appropriate installer for your operating system. Follow the installation instructions provided by the installer to set up Git on your machine.

Git common commands

Some common Git commands include git clone (to clone a repository), git add (to stage changes for commit), git commit (to save changes to the repository), git push (to upload changes to a remote repository), and git pull (to fetch and merge changes from a remote repository).

Issues before containerization

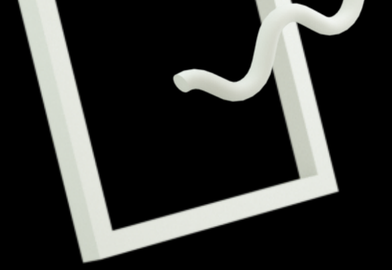
Before containerization, common issues included inconsistent environments between development, testing, and production stages, difficulties in managing application dependencies, and challenges in achieving reproducible and portable deployments.

What is Docker?

Docker is an open-source platform that enables the creation, deployment, and management of containers. Containers provide a lightweight and isolated environment for applications and their dependencies, ensuring consistent and reproducible deployments across different systems.

Docker installation

To install Docker, you can visit the official Docker website (https://www.docker.com/) and download the Docker Desktop application for your operating system. Follow the installation instructions provided by the installer to set up Docker on your machine.



Docker common commands

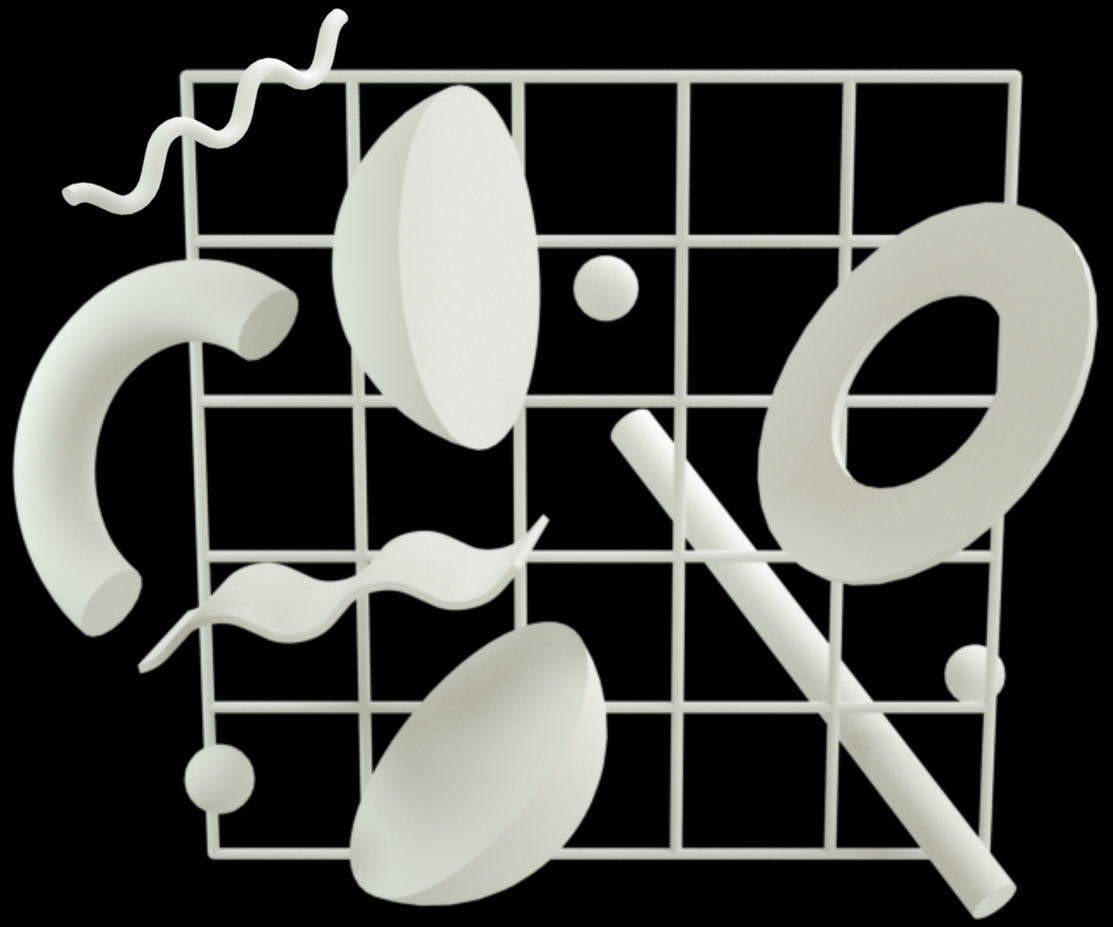
Some common Docker commands include docker run (to start a container), docker stop (to stop a running container), docker build (to build an image from a Dockerfile), docker push (to upload an image to a registry), and docker pull (to download an image from a registry).

Docker environment

Docker provides a containerized environment that isolates applications and their dependencies from the underlying system. It allows for easy replication, scalability, and portability of applications across different environments and systems.

Dockerfile

A Dockerfile is a text file that contains instructions for building a Docker image. It specifies the base image, adds application dependencies, configures the environment, and defines the commands to run when a container is created from the image.

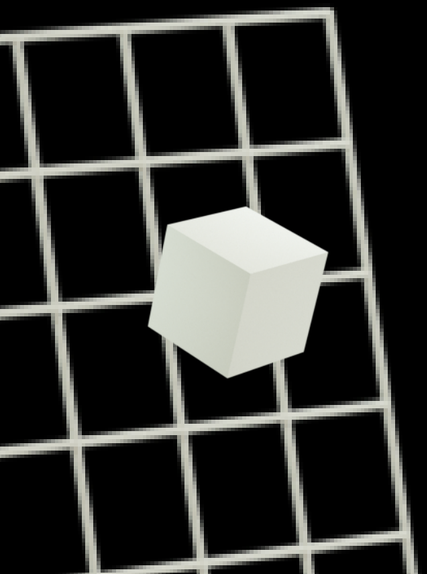


What is Jenkins?

Jenkins is an open-source automation server used for continuous integration, continuous delivery, and continuous deployment (CI/CD). It provides a platform for building, testing, and deploying software projects with extensive plugin support and integration capabilities.

Jenkins overview

Jenkins allows for automated software builds, code testing, and deployment pipelines. It supports integration with version control systems, build tools, and testing frameworks, enabling teams to automate the software delivery process and ensure consistent quality.



Jenkins installation

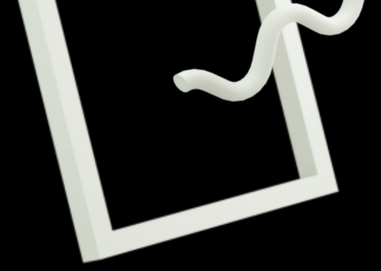
To install Jenkins, you can visit the official Jenkins website (https://www.jenkins.io/) and download the Jenkins package suitable for your operating system. Follow the installation instructions provided to set up Jenkins on your server or local machine.

Continuous monitoring

Continuous monitoring involves the real-time tracking and analysis of system and application metrics to ensure optimal performance, identify issues, and facilitate proactive troubleshooting. It helps in maintaining system availability, performance, and reliability.

Monitoring with ELK

ELK is a popular open-source stack used for log management and analysis. It consists of Elasticsearch for indexing and searching logs, Logstash for log ingestion and processing, and Kibana for visualization and dashboarding. ELK stack can be used for centralized logging and monitoring of applications and infrastructure.



Thx :)